**ML-Agents Toolkit**

Dona totes les eines necessàries per utilitzar Unity com a motor de simulació per aprendre diferents polítiques dels agents posats dintre del motor.

**Parts de ML-Agent Toolkit**

* Espai d’aprenentatge
* Python Low-Level API
* Broquer (Comunicator)
* Python Trainers
* Gym Wrapper (I won’t explain this one)

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

*Espai d’aprenentatge*

Aquí esta l’escena d’Unity i els personatges. Els personatges que per requeriment del problema s’han definits com Agents, podrán observar, prendre desicions i tenir recompenses.

Els Agents de les escenes d’Unity han de tenir mínim tres fitxers ajuntats a ells com a components.

* Classe que modelitza l’aprenentatge del Agent i que hereta de la classe Agent
* Behavior Parameters Class
* Decision Requester

*Agent*

Classe que conte mètodes que poden ser sobre escrits. La seva API contempla mètodes per poder generar observacions del medi, per prendre accions i per assignar recompenses.

La classe Agent conte altres mètodes que poden ser sobre escrits. Cada Agent esta relacionat amb un Behavior Parameter class i a un Decision Requester.

*Behavior Parameter*

Classe que esta pensada per parametritzar i relacionar la classe Agent amb la configuració externa a Unity.

Un Behavior pot ser del següents tres tipus.

* Learning
* Heuristic
* Inference

Si el Behavior del Agent es de tipus Learning, vol dir que hi ha un model i que s’espera entrenar fent us de *mlagents-learn*. Si es de tipus Heuristic, vol dir que el comportament del Agent be donat per l’interacci’o amb un jugador i finalment, si es de tipus inference, vol dir que ja este un model de *Reinforment-Learning* ja entrenat i que l’Agent es comportarà segons aquesta xarxa neuronal.

Els paràmetres importants d’aquesta classe són

* Behavior name (ID únic, serveix per fer referencia a un fitxer de configuaració extern)
* Vector Observations
* Actions
* Model (Xarxa neuronal entrenada)
* Behavior Type (Learning, Heuristic, Inference)
* Use child sensors

*Decision Requester*

Es un Script que serveix per forçar que l’Agent prengui decisions, sense aquest Script linkat com a Component en el nostre Agent, l’Agent mai prendrà decisions. Aquí es pot configurar el període de decisions.

El Behavior Parameter Script i el Decision Requester Script, poden ser linkats a diferents Agents, però un Agent només fa referencia a un Behavior i Decision Requester.

Diagrama

Descripción generada automáticamente

**Utilitzar TensorBoard per fer seguiment de l’entrenament del Agent**

Quan l’entrenament s’està executant, ML-Agents Toolkit guarda estadístiques dintre del directori *results* en un directori amb nom únic, donat a l’hora d’executar el procés *mlagents-learn*.

Per observar el procés d’entrenament quan esta correguent el procés o quan no, ens apropem a la carpeta de resultats i executem: *tensorboard –logdir \*/results${training id}*.

Això obrirà un servidor per servir el contingut estàtic en un dels ports de la teva màquina.

**Arquitectura final**

Diagrama

Descripción generada automáticamente